Learning the Classifier Combination for Image Classification

Deyuan Zhang
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
Email: dyzhang@insun.hit.edu.cn

Bingquan Liu, Chengjie Sun and Xiaolong Wang
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
Email: {liubq, cjsun, wangxl}@insun.hit.edu.cn

***Abstract*—Although some image features and algorithms succeed in many tasks such as scene recognition and face recognition, carefully choosing image features and classifiers are time consuming for a specific image classification task. In this paper, we propose a method that automatically combines the classifiers with probability outputs from different features. We fomulate the problem in quadric programming framework, and solve it efficiently. In addition, we proposed two classifier selection algrithms for selecting the most discriminative classifiers for the speed-accurcy trade-off. The experiment on Corel image dataset shows that our algorithm can fuse the classifiers robustly, and the classifier selection algorithm is flexible and effective.**

***Index Terms*—image classification, classifier combination, classifier selection, max-margin approach, weighted average**

I. INTRODUCTION

Image classification is defined as the labeling of images into one of predefined image labels. It can help users to organize and to browse images, and can also aid in content-based image retrieval since the image's predefined labels can help narrowing the semantic gap between low-level features and high-level semantics [1]. Although this is usually not a very difficult task for humans, it has been proved to be an extremely difficult problem for image classification algorithms [2]. The major problem is the semantic gap between the low-level image features and the conceptual information perceived by humans [3].

Many image classification systems have been developed since the early 1990s, and various image features and classification algorithms are proposed in the literature to tackle some domain-specific image classfication tasks and some benchmark datasets, such as Gist [4] feature for the scene classification, Haar Wavelet [5] feature for the face detection, SIFT [6] feature for the object detection, etc. Various image representations are adopted in these systems: the images are represented by global, block-based, patch-based or region-based local features. Based on the image features, many pattern classification and machine learning alogrithms from different assumptions are adopted for the image classification tasks, such as K-nearest neighbor (KNN) [7], Support Vector Machines (SVM) [8], Hidden Markov Model(HMM) [9], Diverse Density(DD) [10], and so on. While these algorithms suceed in some "benchmark" image datasets, they remained in the naïve way for the image classification task because there is no universal representation for the image understanding as human vision system. Therefore, for a specific dataset, image features and classification algorithms must be chosen carefully, which is a time-consuming work. The robustness of some image classification system could be collapsed when applied to new image databases.

In order to alleviate the situation that the performances of image classification systems depend upon the corresponding image databases, automatically combine classifiers on different image features can be considered as a promising direction. Combination of classifiers from different feature sets becomes more successful due to the fact that classifiers are different and informative [11].

There exist many researches on combining multiple classifiers, and some get excellent performance on some benchmark datasets, but most of these researches mainly focus on the classification accuracy. While for the image classification task, both the accuracy and the speed are important. For a practical image classification system, the structure of the combination method must be considered. Combination methods which are time consuming will not be adopted in the system. In addition, the combination method must be tuned for the speed-accuracy trade off. As we know, combining all the classifiers can gain the highest performance, but the classification speed is the slowest. How to automatically discard some image features for the speed consideration while not losing significant accuracy must be handled in the classifier combination method.

In this paper we focus on the Weightd Average [21] based classifier combination. Some theories try to find the optimal combination method when applying to various classification tasks. According to the previous study [22], Simple Average [24](sum rule) is the optimal linearly combining rule, only if the individual classifiers exhibit both identical performances and correlations between estimation errors; otherwise, Weighted Average can provide better results. For the image classification task, the performance of image features varied too much,

doi:10.4304/jcp.6.8.1756-1763

which support the use of Weight Average in image classification systems. In addition, it is easier to combine some of the most discriminative features than Simple Average.

In order to construct a robust and flexible image classification system, a unified framework is proposed in the paper. The framework automatically learns the weight of the classifiers built on different image features. The problem is formulated in max-margin framework, leading to a quadric programming problem which can be solved efficiently. Classifier selection methods are also proposed in the paper to tackle the speed-accuracy tradeoff of the image classification system. The framework is flexible and easy to tune, which makes our framework more suitable for the image classification task.

The paper is structured as follows: firstly, related works of combining classifiers are described in Section II. In Section III we propose the framework for image classification, including feature extraction, sub-classification, confidence transformation and combination process. Our Max-Margin based Weight Learning algorithm is formulated in Section IV and the corresponding classifier selection methods is described in Section V. Next, a quantitative evaluation of different approaches is shown in Section VI, along with the discussion of the results. A summary and conclusions from this work(Section VII) end this paper.

Your goal is to simulate the usual appearance of papers in a Journal of the Academy Publisher. We are requesting that you follow these guidelines as closely as possible.

II. RELATED WORKS

The idea of combining multiple classifiers has gained a lot of attention by researchers since the combination of classifiers is more accurate than a single classifier.[11] Methods of combining classifiers are roughly categorized into two categories: partition the data into different training sets and form a classifier (Bagging[12] and Boosting[13]); combining the classifiers trained on different classifiers. Because partition the data into different training sets is not related to the paper, we introduce the work of combining different classifiers in detail. Let $C=\{C_1,C_2,\ldots,C_K\}$ be the set of classifiers, and let $L=\{l_1,l_2,\ldots l_J\}$ be the set of class labels. For each input sample x, the classifer C output K decision values corresponding the class labels $f_k(l_i|x)$, as shown in Fig.1.

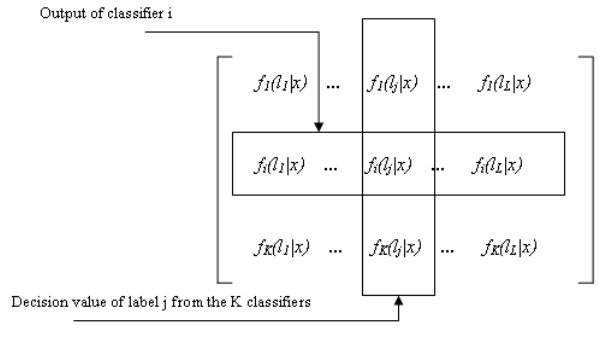

Figure 1. The output of K classifiers on J labels

The problem is how to fuse these decision values to the final decision. The sub-classifiers are different classifer models or trained on various feature sets. For combining the classifiers, according to Xu's [14] methodology, the classification methods are divided into three categories based on the output of the sub-classifiers: the abstract level, the rank level and the measurement level. Many combination rules have been developed on each level. In the abstract level each sub-classifier output a unique label to each class, which is a binary value. The Majority Vote, Naïve Bayes [14], BKS [15] and Wernecke [16] approach are proposed to combine the binary values. Although some successes are made by this approcach, the decision information from sub-classifiers is too little. In the rank level sub-classifiers output the prefered list of each class. Compared to the abstract level, the classifier output more information to the final decision. Researchs based on this level are rare. Borda Count [17] is a simple and fast method based on the rank level. While the rank level output more information than the abstract level, we can only get some information on which class is the one label than the other, but not the confidence of one label exceeds the other one. In the measurement level, confidence value (which is often probability) is addressed by each sub-classifier. The output is a real value, which indicates the confidence of the classifier on the corresponding label. The measurement level contains the highest amount of information among the three levels. Many fusion strategies have been proposed based on this level. Some fixed combination rules have been proposed, as listed below:

The sum rule:

$$l = \arg\max\left\{\sum_{k=1}^{K} f_k(l_t \mid x), l_t \in L\right\} \quad (1)$$

The product rule:

$$l = \arg\max\left\{\prod_{k=1}^{K} f_k(l_t \mid x), l_t \in L\right\} \quad (2)$$

The min rule:

$$l = \arg\max\left\{\min_{k=1}^{K}(f_k(l_t \mid x)), l_t \in L\right\} \quad (3)$$

The max rule:

$$l = \arg\max\left\{\max_{k=1}^{K}(f_k(l_t \mid x)), l_t \in L\right\} \quad (4)$$

Kittler[18] proposed a theoretic framework to interpret these combination rules based on different assumptions. Although these rules are simple, they succeed in many applications and benchmark datasets. In the theory, when applied to independent features, the product rule is more suitable than the sum rule. In image classification, image features tend to be independent to each other, which makes the product rule can achieve better performance than the sum rule. But the product rule is nonlinear, which makes the selection of part of features difficult. The framework proposed by us turn the product rule into linear rule by transforming the probability to another confidence value, which make our framework can get better performance and select classifiers easily.

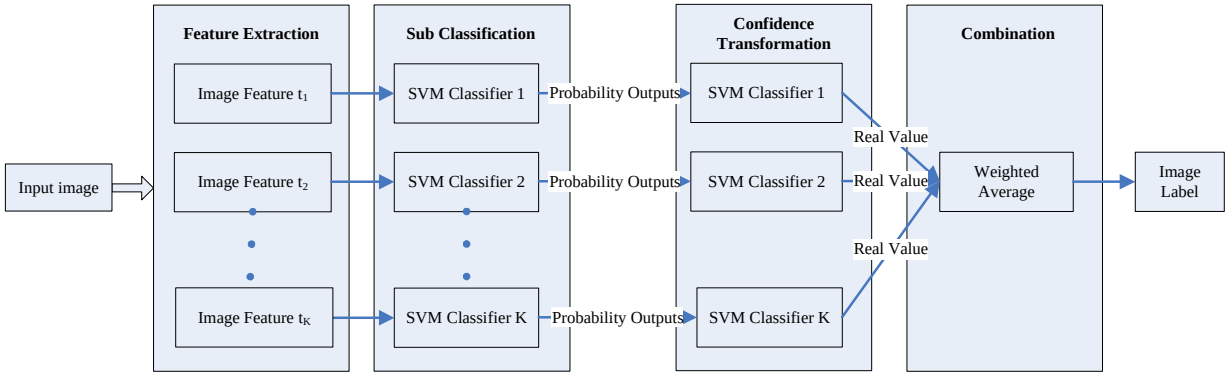

Figure 2. The proposed framework for image classification

There exists other families of works for combining classifiers, such as Fuzzy integral [19], evidential reasoning [20], Decision templates [21]. These works succeed in many fields, but for the image classification task, these methods lose the flexibility of controling the speed-accuracy tradeoff. A huge amount of researchs on classifier combination are proposed in the literature, and we can not fully explore all of them. Please review [25] for a comprehensive study.

III. Framework Description and Problem Formulation

A. Framework Description

The proposed framework is described in Fig.2. Assume there are J predefined labels or categories, and $l_j(j=1,\ldots,J)$ denotes the jth image labels. Given the input image x, the image classification task is to assign a predefined image label l_j to the image. As Fig.2 shows, the proposed framework is composed of four processes: feature extraction, sub-classification, confidence transformation and automatic combination. For the input image x, during the feature extraction process various image features are extracted from the input image, each of which is denoted by $t_i(i=1,\ldots,K)$. Let T= $\{t_1, t_2,\ldots,t_K\}$ denotes the whole image feature set. In the process of sub-classification classifiers which output the probabilities of class labels are built on the corresponding image features. Let $f_k(l_j|x)$ is the probability that the input image x belongs to the jth label predicted by the classifier built on the feature t_j. In the process the confidence transformation, the output porbabilitie $f_k(l_j|x)$ is transforemed to another confidence value $F_k(l_j|x)$ which is used to improve the performance of the classifier combination methods. In the combination process the outputs are incorporated by weighted combination to gain the final class label.

B. Support Vector Machines

Support Vector Machines (SVM) are a set of supervised learning algorithms first introduced by Vapnik [26], and succeed in many classification tasks. Given a training set of instance-label pairs (x_i, y_i), i = 1,2,……,l, the SVM require the solution of the following optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2} w^T w + C\sum_{i=1}^{l} \varepsilon_i \\ s.t. \quad & \\ & y_i(w^T \varphi(x_i)+b) \geq 1-\varepsilon_i \\ & \varepsilon_i \geq 0 \end{aligned} \tag{5}$$

Here training vectors x_i are mapped into a higher dimensional space by the function Φ. To classify a sample x, the decision function is given by:

$$f(x) = sign(\sum \alpha_i K(x_i, x) + b) \tag{6}$$

Where $K(x_i,x)= \Phi(x_i)\ \Phi(x)$is the kernel function.

In order to obtain the likelihood for a test sample to be in each predefined category, the outputs (decision values) from each corresponding SVM will be mapped into probabilities by training the parameters of a sigmoid function [27] which is defined as:

$$P(y=1|\ f) = \frac{1}{1+\exp(Af+B)} \tag{7}$$

Where P is the probability, y is the actual label, f denote the output from SVM, and A and B are the two parameters estimated from a set of (f_i, y_i)s [27].

The two class classification problem can be adapted to multi-class classification task by one-against-one approach proposed by Wu [28], which calculates the probability output of each binary classifier.

C. Confidence Transformation

Given the probability outputs of SVM for the image labels, we transform the probability to the confidence value:

$$F_k(l_j \mid x) = \log(f_k(l_j \mid x)) \tag{8}$$

By this confidence tranformation, the product rule can be rewrited as:

$$l = \arg\max\left\{\sum_{k=1}^{K} F_k(l_t \mid x), l_t \in L\right\} \tag{9}$$

Through this confidence transformation, the sum and product rule can be processed in the same framework. The confidence transformation procedure is optimal, but when applying to the image classification tasks, theoritical results [24] indicate that the product rule fits better than the sum rule. We show in the experiment section that the transformation procedure can gain an increasement of the fusion method's performance.

IV. Max-Margin based Weight Learning

In this section we describe our Max-Margin based Weight Learning (MMWL) approach in detail. Based on the framework proposed above, the task is to learn the weight of each sub-classifier:

$$l = \arg\max\left\{\sum_{k=1}^{K} w_k f_k(l_t \mid x), l_t \in L\right\} \qquad w_k \geq 0 \quad (10)$$

It is a Weighted Average approach. Inspired by the large margin framework, we propose a learning scheme to determine the weight of classifiers. Given a training image set $x_i(i=1,2,\ldots,M)$and corresponding label y_i, and in the ideal case, for all the training images, the final decision should classify the training image exactly. That is, for all the training images, the weight w_k of each classifier C_k will satisfy the following equation:

$$\sum_{k=1}^{K} w_k f_k(y_i \mid x_i) = \max_{l=1}^{L} \sum_{k=1}^{K} w_k f_k(l \mid x_i) \qquad w_k \geq 0 \quad (11)$$

That is, for each training image x_i, the final decision value of label y_i is larger than the value of label l if label l and y_i are not identical, resulting in the following inequation constraint hold:

$$\sum_{k=1}^{K} w_k f_k(y_i \mid x_i) - \sum_{k=1}^{K} w_k f_k(l_t \mid x_i) > 0 \qquad (y_i \neq l) \quad (12)$$

Denote $f(l|x_i)$ is a K dimensional column vector of the decision value of label l by the K sub-classifiers for image x_i, and w is the column vector of the weight of the K sub classifiers, the constraints can be rewritten as:

$$w^T \bullet f(y_i \mid x_i) - w^T \bullet f(l \mid x_i) > 0 \qquad (y_i \neq l) \quad (13)$$

By scaling the weight vector w, the following inequation can always holds:

$$w^T \bullet f(y_i \mid x_i) - w^T \bullet f(l \mid x_i) \geq 1 \qquad (y_i \neq l) \quad (14)$$

While for the real image classification problems, the constraints can not be satisfied for all the images. To overcome this problem, we add hinge-loss on the constraint:

$$w^T \bullet f(y_i \mid x_i) - w^T \bullet f(l \mid x_i) \geq 1 - \xi_{il} \qquad (y_i \neq l, \xi_{il} \geq 0) \quad (15)$$

In order to prevent overfitting, we add a L2 regulization penalty on w. Thus, the objective function is defined as:

$$\min \quad \sum_{i=1}^{M} \sum_{l \neq y_i} \xi_{il} + \lambda \|w\|^2$$

s.t.

$$w^T \bullet \left(f(y_i \mid x_i) - f(l \mid x_i) \right) \geq 1 - \xi_{il} \quad (16)$$
$$\xi_{il} \geq 0$$
$$w_t \geq 0$$

Let the $C^* = 1/2\lambda$, the objective function is defined as:

$$\min \quad \frac{1}{2}\|w\|^2 + C^* \sum_{i=1}^{M} \sum_{l \neq y_i} \xi_{il}$$

s.t.

$$w^T \bullet \left(f(y_i \mid x_i) - f(l \mid x_i) \right) \geq 1 - \xi_{il} \quad (17)$$
$$\xi_{il} \geq 0$$
$$w_t \geq 0$$

The objective function is a convex programming problem which can be solved efficiently.

V. CLASSIFIER SELECTION

Combining all sub-classifiers could bring precise results, but executing all the classifiers is time consumming. In practice, most of the image classification system is spend on the feature extraction process, the classier selection technique could make some features do not need be extracted, resulting in increasing the speed of the system. In addition, because the performance of image feautres is not identical when applying real image datasets and some classifiers contribute no or a little to the final decision, discarding these features and the corresponding classifiers will not degrade the performance too much. Therefore, discarding some classifiers that comtribute less to the final decision is appropriate for improving the speed of the system. In the framework proposed in the paper, the classifer is "less" contribute if is weight is small. If it is 0, the classifer do not contribute the performance, and can be discarded without the decrease of performance. We use an iterative method for the classifier selection. That is, in every iteration, we remove the classifer which have the minimum weight, and retrain the weight of other classifiers using MMWL algorithm. The algorithm stops when some stop conditions are satisfied. The algorithm is described in Fig.3. Although the solution is not garanteed to be the optimal combination, this greedy approach is fast and works well in practice.

For the image classification system two ways for selecting classifiers are natural: one is to select and combine N classifiers for obtaining the best performance; the other is to discard as many classifiers as possible while preserving the performance of the system. These two methods focus on the performance and speed seperately.

These two classifier selection methods depend on the function CHECK_STOP(w) of the algorithm. For the first method, the function CHECK_STOP(w) is simple. That is checking if the number of classifers exceeds the predefined N. We call this methd N-Select. For the second method, we use a parameter λ to control if the algorithm stopped:

$$\forall w_i \qquad w_i \geq \lambda \bullet average_{t=1}^{|Label|}(w_t) \quad (18)$$

```
SELECTION{
    INPUT decision value f_k(l|x_i) as H
    Label = {1,2,…,L}
    w = MMWL(H)

    while(CHECK_STOP(w)){
        t = argmin{ w_i, i Label}
        remove t from Label
        remove f_t(l|x_i) from H
        w = MMWL(H)
    }
}
```

Figure 3. The framework of classifier selection algorithm

Where w_i is the ith dimension of the weight vector w.The intuition is that if the weight w_i of every classifer C_i contributes more than λ*average(w), we think the every classifier is important for the final decision. This method is called λ-Select.

VI. EXPERIMENT

A. Experimental setup

In order to test the performance of the proposed algorithm, we use the Corel image dataset [2] for evaluation. The dataset contains ten categories of images chosen from Corel Stock Photo CD: Africa, Beach, Building, Buses, Dinosaur, Elephant, Flower, Horse Mountain and Food which are represented as Category 0 to 9. The example image of each category is listed in Fig.4. Some categories (for example "Dinosaur") are easy to classify while some categories are difficult(such as category "Beach" and "Mountain"). Each category contains 100 images. Seven image features are used in the proposed framework: Color Histogram [29] in RGB(CH-RGB), LUV(CH-LUV) and Lab(CH-Lab) color space, Color Coherence Vectors (CCV) [30], Edge Histogram (EH) [31], Scalable Color descriptor (SC) [31] and Pyramid-structured Wavelet Transform [32](PWT). These image feature is easy to compute and can be computed in less than 1 second.

For training the SVM, we use the libsvm [33] toolbox and adopt the linear kernel. Although other kernels perhaps perform better results, the linear kernel is aproporate for high dimensional image features and relatvie fast speed. The parameter C of SVM is set to 10. In the dataset all the features performs well for the experimental setup.

We repeat each experiment for 10 random splits of train and test images, and report the average accuracy of the results obtained over 10 different test sets.

Our WWML based methods are compared in the following experiment, both with (denoted as LOG-MMWL) and without (denoted as MMWL) confidence transformation. Analogy to the sum and product rule which is used under different assumptions, we show that LOG_MMWL and MMWL have similar behaviours. The parameter of our algorithm C^* is set to 0.1 for all experiments.

B. MMWL results

We compare our MMWL learning method with some fixed rule methods: the average, product, max, min combination methods, and the best sub-classifier Scalable Color descriptor(SC). Firstly, we compare the performance of 50 training images in each category. The average accuracy is reported in Table.I

Figure 4. Sample images taken from 10 categories

TABLE I. THE AVERAGE ACCURACY OF MMWL , LOG-MMWL MEHTODS, PRODUCT, SUM, MIN, MAX RULE METHODS, AND THE BEST SINGLE FEATURE SC

Method	Accuracy
MMWL	0.899
LOG-MMWL	0.9116
PRODUCT	0.8836
SUM	0.8712
MIN	0.8696
MAX	0.8392
SC	0.845

As shown in the Table I, Both the MMWL and LOG-MMWL achieves better than other fusion rules. In addition, the product methods are better than the sum methods, and LOG-MMWL performs better than MMWL methods, which confirms that the image features are more independent. To compare the Simple Average and Weighted Average, the MMWL outperforms the SUM rules by 2.78%, and LOG-MMWL outperforms the PRODUCT rule by 2.8%. Compared to the state of the arts, our fusion stratigey is comparable to the state of the art method of this dataset, our LOG-MMWL outperforms Qi's [1] fusion method by 2.36%, although our sub-classifier's performance is degrade to the Qi's sub-classifiers.

Secondly, we make a closer analysis of the performance by looking at classification results on each category in terms of the confusion matrix. The results are listed in Table II(MMWL) and Table III(LOG-MMWL) seperately. The numbers on the diagonal show the classification accuracy for each category, and off-diagonal entries indicate classification errors [2]. As Table II and Table III shows, there exist some similarities in the confusion matrix. There are some classes which are easy to be classified, such as Dinosaur and Horse. The largest errors between categories are errors between Category 1(Beach) and Category 8(Mountain). Nearly more than 10% of the images in these categories are misclassified each other. The misclassification is mainly caused by the fact that "Beach" and "Mountain" categories are both semantically and visually similar.

Thirdly, In order to test the the sensitivity of the proposed method to the size of training images, we random select 10, 20, 30, 40, 50 images in each category and use the other images to test. Fig.5 gives the classification results compared with the other four fusion methods and the performance of sub-classifiers.

As Fig.5 shows, both of our methods nearly achieve the best performance in every setup, except the performance of 10 training images. This is mainly because 10 training images in every category is insufficient for estimating the probability of each SVM classifier, leading to some overfitting of the weight of the classifiers. In every setup MMWL and LOG-MMWL outperforms the sub-classifiers by nearly 10%, which comfirms the effectiveness of our algorithm.

TABLE II. THE CONFUSION MATRIX OF MMWL IMAGE CLASSFICATION RESULTS(OVER 10 RANDOM SPLIT TRAIN AND TEST IMAGES)

	Cat. 0	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8	Cat. 9
Cat. 0	84.20	2.20	4.40	0.40	0.80	5.00	0.20	0.60	0.00	2.20
Cat. 1	2.60	71.60	4.20	4.00	0.00	1.80	0.00	0.80	13.40	1.60
Cat. 2	1.40	3.80	85.60	0.80	0.60	1.00	0.60	0.40	4.60	1.20
Cat. 3	0.40	0.20	0.80	98.40	0.00	0.00	0.00	0.00	0.00	0.20
Cat. 4	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
Cat. 5	3.60	1.00	0.60	0.00	0.00	89.00	0.00	2.60	3.00	0.20
Cat. 6	0.80	0.20	0.00	0.80	0.20	0.00	97.00	0.80	0.00	0.20
Cat. 7	0.00	0.20	0.00	0.00	0.00	0.80	0.00	99.00	0.00	0.00
Cat. 8	0.00	11.00	4.40	0.00	1.00	1.80	1.00	0.00	80.80	0.00
Cat. 9	2.40	1.20	0.00	0.60	0.00	0.00	1.40	0.40	0.60	93.40

TABLE III. THE CONFUSION MATRIX OF LOG-MMWL IMAGE CLASSIFICATION RESULTS(OVER 10 RANDOM SPLIT TRAIN AND TEST IMAGES)

	Cat. 0	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8	Cat. 9
Cat. 0	84.60	3.20	2.60	0.00	0.80	5.80	0.00	0.00	0.40	2.60
Cat. 1	2.80	75.60	4.40	1.40	0.00	1.20	0.00	0.20	13.20	1.20
Cat. 2	0.60	4.60	87.00	0.00	0.00	0.60	1.20	0.00	4.40	1.60
Cat. 3	0.20	0.80	0.60	98.20	0.00	0.20	0.00	0.00	0.00	0.00
Cat. 4	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
Cat. 5	3.00	1.20	0.20	0.00	0.00	89.80	0.00	1.80	3.60	0.40
Cat. 6	0.00	0.00	0.00	0.00	0.00	0.00	99.20	0.00	0.00	0.80
Cat. 7	0.00	0.00	0.00	0.00	0.00	1.00	0.00	99.00	0.00	0.00
Cat. 8	0.00	9.80	3.60	0.00	0.40	1.80	0.80	0.00	83.60	0.00
Cat. 9	2.20	0.60	0.00	0.00	0.00	0.40	1.00	0.00	1.20	94.60

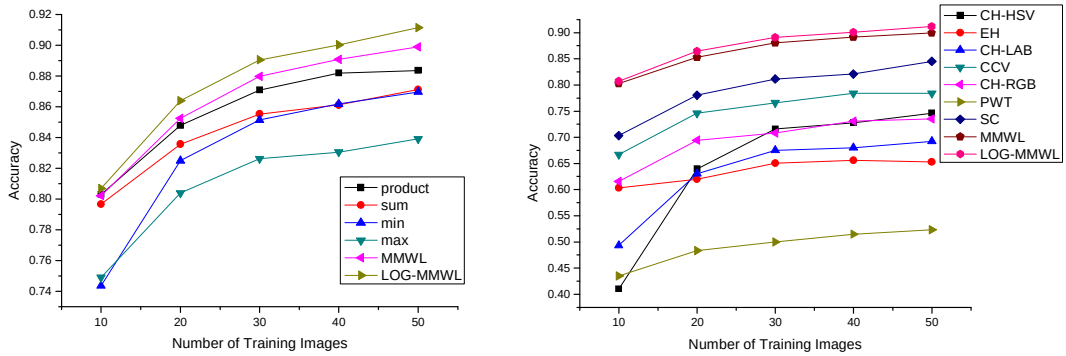

Figure 5. the performance of the MMWL compared to fixed fusion rules and sub-classifiers. Left: performance comparison of MMWL, LOG-MMWL, and the other four fixed rule fusion methods; Right: performance comparision of MMWL, LOG-MMWL with that of the seven sub-classifiers

C. The effectiveness of classifier selection methods

In order to test the effectiveness of the classifier selection methods, we alternate the parameter of the classifier selection algorithms and compare the results of different parameters. Firstly, we test the effectiveness of N-Select method. The N=1, 2, 3 is compared in the paper, and when N>3, the performance is nearly close to combining all the classifiers. The result is showed in Fig.6. As expected, when N increases(selecting more features), the corresponding classification accuracy increases. When N>1, the performance is much superior to that of the best sub-classifier SC. The exception is N=1, our selection algorithm fails to select the most discriminative methods. Although MMWL algorithm is degrade to the LOG-MMWL algorithm, its performance is more robust to the parameter. Fig.7 shows the result of λ-Select method. λ is chosen from 0.6 to 0.9 with a step of 0.1. We do not try λ<0.6 because our algorithm generates sparse weights; the other parameters have similar performance with the MMWL and LOG-MMWL methods separately. The λ controls the the sparsness of the weights, and the algorithm with the larger lamda will generate the less sparse weight. When λ decreases, the performance increases as expected. From Fig.7 we can see that although MMWL has less performance than LOG-MMWL, the MMWL is more robust than the LOG-MMWL method. When λ=0.9 in LOG-MMWL method,

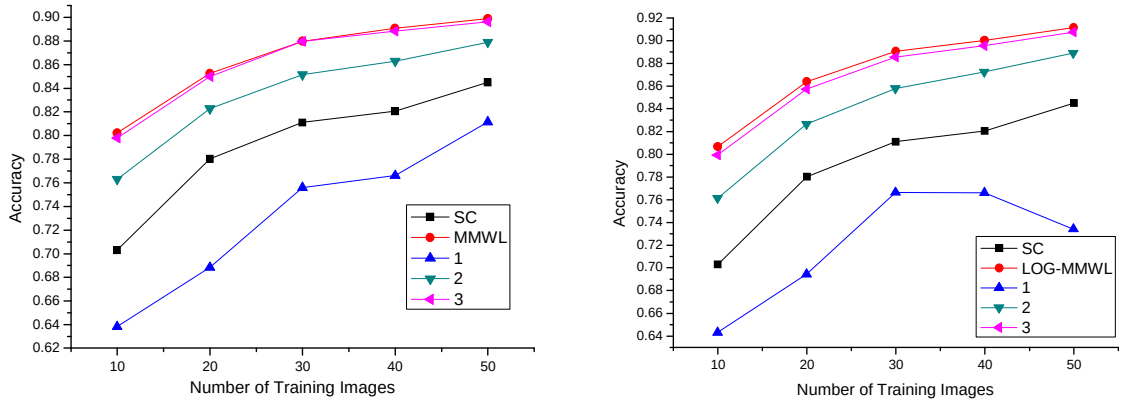

Figure 6. The classification accuracy of the N-Select method. Left: MMWL method; Right: LOG-MMWL method

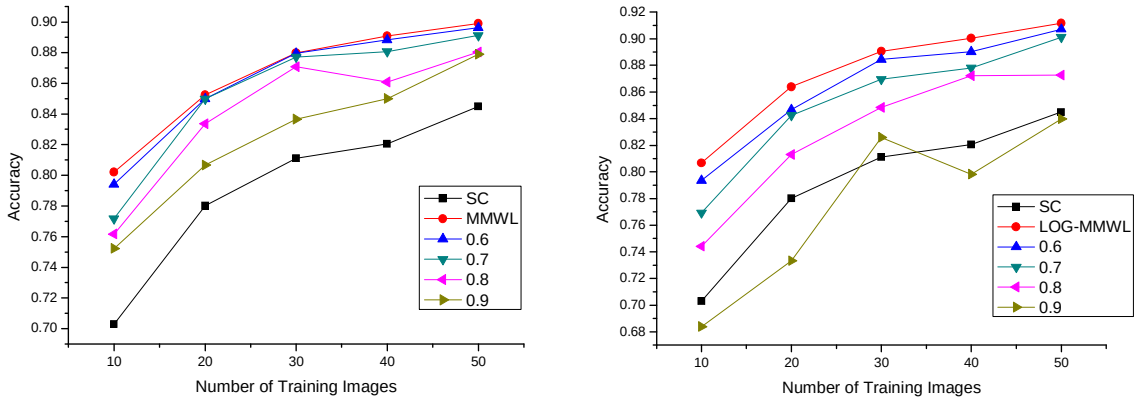

Figure 7. The classification accuracy of the λ -Select method. Left:MMWL method; Right LOG-MMWL method

the performance is degrade to the SC methods. When λ=0.9, in most cases, there is one classifier selected, leading to the same problem with N-Select method through weight learning. Although our method fails to select the best one classifier, our methods succeed in selecting two or more classifiers.

VII. Conclusion

In this paper, we advocate the weighted average framework, and propose a unified framework that combined multiple image features. Based on these classifiers computed on multiple image features, a Max Margin based Weight Learning method is proposed to learn the optimal weight of classifiers. Based on the MMWL method, we also proposed a classifier selection method to gain the speed-accuracy trade-off. Experimental results show that our MMWL method and the classifier selection method are effective and robust.

Acknowledgment

This investigation was supported by the project of the National Natural Science Foundation of China (grants No. 60973076).

References

[1] X. Qi and Y. Han, “Incorporating multiple SVMs for automatic image annotation”, Pattern Recognition, vol. 40, pp. 728-741, 2007.

[2] Y. Chen and J.Z. Wang, “Image Categorization by Learning and Reasoning with Regions”, Journal of Machine Learning Research, vol. 5, pp. 913-939, 2004.

[3] J.Z. Wang, J. Li, and G. Wiederhold, “SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 947-963, 2001.

[4] A. Oliva and A. Torralba, “Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope,” International Journal of Computer Vision, vol. 42, May. 2001, pp. 145-175.

[5] P. Viola and M.J. Jones, “Robust Real-Time Face Detection,” Int. J. Comput. Vision, vol. 57, 2004, pp. 137-154.

[6] D.G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” Int. J. Comput. Vision, vol. 60, 2004, pp. 91-110.

[7] M. Szummer and R.W. Picard, “Indoor-Outdoor Image Classification,” IEEE International Workshop on Content-based Access of Image and Video Databases, in conjunction with ICCV'98, pp. 42-51, 1998.

[8] O. Chapelle, P. Haffner, and V. Vapnik, “Support vector machines for histogram-based image classification”, IEEE Transactions on Neural Networks, vol. 10, pp. 1055-1064, 1999.

[9] J. Li and J. Wang, "Automatic Linguistic Indexing of Pictures by a statistical modeling approach", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, pp. 1075-1088, 2003.
[10] O. Maron and A.L. Ratan, "Multiple-Instance Learning for Natural Scene Classification", Proceedings of the Fifteenth International Conference on Machine Learning, pp. 341-349, 1998.
[11] R. Duin, "The combining classifier: to train or not to train?," Pattern Recognition, 2002. Proceedings. 16th International Conference on, 2002, pp. 765-770 vol.2.
[12] L. Breiman, "Bagging Predictors," Machine Learning, 1996, pp. 140-123.
[13] Y. Freund and R. Schapire, "Experiments with a New Boosting Algorithm," In Proceedings of the Thirteenth International Conference on Machine Learning, 1996, pp. 156-148.
[14] L. Xu, A. Krzyzak, and C.Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," Systems, Man and Cybernetics, IEEE Transactions on, vol. 22, Jun. 1992, pp. 418 -435.
[15] Y.S. Huang and C.Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, 1995, pp. 90-94.
[16] K.D. Wernecke, "A coupling procedure for the discrimination of mixed data," Biometrics, vol. 48, Jun. 1992, pp. 497-506.
[17] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems," IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, 1994, pp. 66-75.
[18] J. Kittler, M. Hatef, R.P. Duin, and J. Matas, "On Combining Classifiers," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, 1998, pp. 226-239.
[19] M. Sugeno, "Fuzzy measure and fuzzy integrals, a survey", Fuzzy Automata and Decision Processes. pp. 89-102, 1997
[20] Y. Bi, J. Guan, and D. Bell, "The combination of multiple classifiers using an evidential reasoning approach," Artificial Intelligence, vol. 172, Oct. 2008, pp. 1731-1751.
[21] C. Liu, "Classifier combination based on confidence transformation," Pattern Recognition, vol. 38, Jan. 2005, pp. 11-28.
[22] K. Tumer and J. Ghosh, "Error Correlation and Error Reduction in Ensemble Classifiers," Connection Science, vol. 8, 1996, pp. 385-404.
[23] F. Li and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 524-531, 2005.
[24] D.M.J. Tax, M. van Breukelen, R.P.W. Duin, and Josef Kittler, "Combining multiple classifiers by averaging or by multiplying?," Pattern Recognition, vol. 33, Sep. 2000, pp. 1475-1485.
[25] L.I. Kuncheva, "Combining classifiers: Soft computing solutions," PATTERN RECOGNITION: FROM CLASSICAL TO MODERN APPROACHES, vol. 2001, 2001, pp. 427--452.
[26] C. Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, vol. 20, pp. 273-297, 1995.
[27] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods", Advances in Large Margin Classifiers, pp. 61-74, MIT Press, 1999.
[28] T.F. Wu, C.J. Lin, and R. C. Weng. "Probability estimates for multi-class classification by pairwise coupling", Journal of Machine Learning Research, vol. 5, 975–1005, 2004.
[29] M.J. Swain and D.H. Ballard, "Color indexing," International J Computer Vision, vol. 7, pp. 11-32, 1991G. Pass, R. Zabih, and J. Miller, "Comparing Images Using Color Coherence Vectors," ACM Multimedia, pp. 65-73, 1996.
[30] G. Pass, R. Zabih, and J. Miller, "Comparing Images Using Color Coherence Vectors," ACM Multimedia, pp. 65-73, 1996.
[31] B. Manjunath et al., "Color and texture descriptors," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 11, pp. 703-715, 2001.
[32] M.F.A. Fauzi and P.H. Lewis, "Texture-based image retrieval using multiscale subimage matching," Proceedings of the SPIE on Image and Video Communications and Processing, vol. 5022, pp. 407-416, May. 2003.
[33] C. C. Chang and C. J. Lin. "LIBSVM: a library for support vector machines", 2001.